

## リサーチデザインと言語統計処理（2015 年度前期木曜 5 時限目）

・参考文献：青木繁伸『R による統計解析』

### ■ R による統計処理入門・統計分析例（担当：杉浦）

1. 7 月 9 日 R による統計処理入門（R の基本的な使い方の説明）
  2. 7 月 16 日 R による統計分析例（分散分析と具体的な分析事例の紹介）前半
  3. 7 月 23 日 R による統計分析例（分散分析と具体的な分析事例の紹介）後半
- 

## 1. R による統計処理入門

WikiPedia

R 言語（アールげんご）は、オープンソースでフリーソフトウェアの統計解析向けプログラミング言語、及びその開発実行環境である。

### 1.1 はじめに

#### ■ インストール

- ・ RipWiki を検索 R のインストール
- ・ <http://www.r-project.org/>
- ・（RStudio もインストールすると便利）

#### ■ 起動と終了

起動：「スタート」から選ぶ

終了：「メニュー」の「ファイル」から「終了」を選ぶか、コマンド

（作業スペースの保存 => パソコン室での作業の場合、自分の USB メモリーに）

#### ■ コンソールでコマンド

「R Console」という窓

プロンプト の右にコマンド（命令）を書く

コマンドを組み合わせ、ひとまとめにしてプログラムにする（「関数」とよぶ）

#### ■ データを「処理」する

統計処理

テキスト処理

処理する前に、データを整形する

データは、文字か数字

数字のデータを整形する

エクセルなどで、プログラムの「想定している」形式に合わせる。

文字のデータを整形する

コーパス・データのフォーマット

## CHILDES の CHAT フォーマットなど

### ■習得しないといけないこと

自分に必要なコマンド・プログラムは何か

## 1.2 作業ディレクトリー、作業スペース ( ) そして、履歴 ( )

### ■作業ディレクトリー

- ・パソコンの「ファイルシステム」の中のどこで作業をするか
- ・どのフォルダー内にあるデータを処理するか
- ・処理するデータのあるフォルダーへ移動しておく
- ・メニュー「ファイル」=>「」
  - ・USB メモリー内に授業用のフォルダーを作っておくとよい
- ・移動して、今どこにいるか、確認するには と打ってみる。

```
> getwd()  
[1] "C:/Users/sugiura/Documents"
```

### ■作業スペース ( ) (「ワークスペース」と呼ばれることもある)

- ・R で作業した「内容」(読み込んだり処理して「変数」に入れたデータ等)のすべて
- ・作業ディレクトリー内に自動的に保存される。
- ・終了するときに「作業スペースを保存しますか?」と聞かれる。「はい」と答えると、今いる作業ディレクトリー内に保存される。
- ・ファイル名は「」(拡張子を表示しないようにしていると、表面上は「見えない」)
- ・作業履歴も自動的に同じ場所というファイル名で保存される。

### ■作業を再開するとき

- ・作業ディレクトリーへ移動する。
  - ・メニューの「ファイル」=>「ディレクトリーの変更」
- ・試みて、以前の作業スペースが復元されていることを確認する。
- ・自動的に復元されていない場合は、メニューの「ファイル」から明示的に「作業スペースの読み込み」をする。
- ・「履歴」も同様。矢印キー「↑」を押してみて、以前打ち込んだコマンドが出てくるか確認する。
  - ・出てこない場合は、メニューから読み込む。

## 1.3 R でのデータの取り扱い

### ■変数と代入

- ・変数という「入れ物」に「値」を代入する
- ・変数は、半角英数文字でつける(数字で始めてはいけない。する。)
- ・変数への「代入」は、記号であらわす。

```
seiseki <- 50
```

- ・「seiseki」という変数に数字「50」が入った。

```
naeae <- "Hanako"
```

- ・「namae」という変数に文字「Hanako」が入った。(文字はダブルクォートの記号「」で囲む)

```
> abc <- 3
```

```
> abc # abc の中身を表示させる。
```

```
[1] 3 # 3 が入っている。
```

```
> abc <- 300 # abc の値に 300 が上書きされる。
```

```
> abc # abc の中身を表示。
```

```
[1] 300 # 中身が 300 に変わった。
```

```
> efj <- 6
```

```
> jke <- 2987
```

```
> ls() # これまでに作った変数一覧を表示。
```

```
[1] "abc" "efj" "jke"
```

```
> rm(jke) # 変数 jke を削除。
```

```
> ls() [1] "abc" "efj"
```

- ・文字と数字を区別する

```
> namae <- "sugiura" # namae には sugiura という文字列が入る。
```

```
> class(abc) # class( 変数名 ) で、変数に入っているものが数字なら numeric を、
```

```
# 文字なら character を返す。
```

```
[1] "numeric"
```

```
> class(namae)
```

```
[1] "character"
```

---

## ■ 配列 ( R では「ベクトル」と呼ぶ )

---

- ・配列 = 入れ物が複数ならんでいる変数

```
<- c( , , , ) で入れる
```

```
> kazu <- c(2,4,6,8)
```

```
> kazu # kazu の中身を表示
```

```
[1] 2 4 6 8
```

```
> names <- c("I","you","he") # 配列の中身は文字列でもいい。
```

```
> names [1] "I" "you" "he"
```

```
> length(names) #length( 配列名 ) で、該当配列の要素数を返す。
```

```
[1] 3 # 配列 names の要素は、 "I" "you" "he" の 3 つ。
```

- ・変数や配列に何があるか調べる

```
ls()
```

- ・要素がいくつ入っているか調べる

```
length( 配列名 )
```

## 1.4 R で扱うデータの種類

### ■ ベクトル

- ・プログラミングで言う「配列」
- ・「変数名」をつける
- ・要素が入っている
- ・変数名 <- c( 要素をカンマで区切って並べる )
  - ・例：国語と数学の得点
    - ・kokugo <- c(83, 45, 73, 50, 22, 67, 77, 89, 66, 90, 67, 76, 67, 89, 90, 95, 71, 60)
    - ・suugaku <- c(90, 55, 90, 43, 33, 55, 48, 98, 56, 75, 77, 80, 87, 76, 56, 81, 45, 70)
- ・要素は前から順番に位置が決まっている
- ・変数名をタイプすると内容が表示される

```
> kokugo  
[1] 83 45 73 50 22 67 77 89 66 90 67 76 67 89 90 95 71 60
```

### ■ データフレーム

- ・プログラミングで言う「多次元配列」
- ・行が項目
- ・列が測定値
- ・複数のベクトルを合わせてデータフレームを作ることできる
- ・変数名 <- data.frame( ベクトル名をカンマで区切って並べる )
  - ・例：期末試験の得点（国語と数学）
    - ・kimatsu <- data.frame(kokugo, suugaku)
- ・変数名をタイプすると内容が表示される

```
>kimatsu
  kokugo suugaku
1      83      90
2      45      55
3      73      90
4      50      43
5      22      33
6      67      55
7      77      48
8      89      98
9      66      56
10     90      75
11     67      77
12     76      80
13     67      87
14     89      76
15     90      56
16     95      81
17     71      45
18     60      70
```

- ・ 個々のベクトル（データの縦のカラム）を指定するときは \$ を使う
  - ・ 国語の得点 kimatsu\$kokugo
  - ・ 数学の得点 kimatsu\$suugaku

- ・ データを取り出すこともできる

```
A <- classes$classA
B <- classes$classB
```

## 1.5 データを見してみる

- ・ いくつ要素があるか見る

```
length(x)
```

- ・ 平均を出す

```
mean(x)
```

- ・ レンジ（範囲）を出す

```
range(x)
```

- ・ 標準偏差を出す

```
sd(x)
```

- ・ データの全体像を見る

```
summary(x)
```

### ■ グラフにしてみる

#### 資料のちらばりを表す

- ・ 散布図 `plot()`
  - ・ これだとバラバラなので、並べ替えてみる `sort()`
  - ・ `plot(sort())`
- ・ 二次元 `plot(x,y)`
- ・ 棒グラフ `barplot()`
- ・ ヒストグラム `hist()`
- ・ `stem()`
  - ・ `stem( データ , 2 )` とオプション 2 を付けること
  - ・ 縦棒の左側が 1 0 ( 以上 ) のけた、右側が 1 のけた
- ・ 箱ヒゲ図 `boxplot()`
  - ・ ヒゲの長さは箱の 1.5 倍

### 図を順にかさねていくわざ

## 1.6 事前にデータを作っておく

- ・ 表計算ソフトなどで、テキストファイル ( ) で保存しておく。
- ・ 一件一行
- ・ 一番上の行は、変数名
- ・ 欠損値は NA と記入しておく
- ・ ファイルの読み込み
  - ・ データフレーム名 <- `read.table(choose.files())`
  - ・ これでウインドウが開くのでどのファイルを読み込むか指定する

### ■ エクセルから「コピペ」することもできる

データフレーム名 <- `read.table("clipboard", header=T)`

- ・ カラムの見出し ( ヘッダー ) がいない場合は、のオプションはつけない
  - ・ コピペのタイミングに注意 ↓
1. R のコンソールで、読み込むコマンドを書いておくのだけれど、「Enter」キーを押さないでおく
    - ・ 命令は書いてあるけど、まだ実行されていない状態
  2. エクセルの方で、該当するデータのエリアを選択して、コピーする ( Ctrl+C )
    - ・ これで「クリップボード」にデータが記憶された状態になる
  3. R のコンソールに戻って、「Enter」キーを押す
    - ・ これで命令が実行され、「クリップボード」内のデータが読み込まれる

## 1.7 いろいろな統計処理

### ■ t 検定 ( 平均の差の検定 )

`t.test(x,y)`

- ・ 「対応なし」がデフォルト
- ・ 「対応あり」は、オプションをつける。

```
t.test(x,y, paired=T)
「平均は等しい」(帰無仮説)
その起きる確率がどのくらいあるか?
5%未満、、平均が等しいとはえない(対立仮説)
```

## ■ 正規性の検定 (Shapiro-Wilk 検定)

(「正規分布している」が帰無仮説。つまり  $p > 0.05$  なら帰無仮説を捨てない)

```
shapiro.test(x)
```

## ■ 等分散の検定 (F test)

## ■ Wilcoxon の順位和検定 (正規性のないデータ)

```
wilcox.test(x,y)
```

## ■ 相関係数

```
cor(x,y)
```

## ■ 相関係数の検定

```
cor.test(x,y)
```

## ■ $\chi^2$ 自乗検定 ( $\chi^2$ 二乗検定)

```
chisq.test()
```

二次元のデータの場合、行列 (matrix) でデータを読み込んでおく  

```
sample.data <- matrix(c(976, 89918, 916, 77713), ncol=2, nrow=2)
```

## ■ $G^2$ 検定 (対数尤度比検定)

- ・事前に青木先生のページより関数を source で読み込んでおく。

```
source("http://aoki2.si.gunma-u.ac.jp/R/src/G2.R", encoding="euc-jp")
G2()
```

## ■ 回帰式をプロットする

```
abline(lm(y~x))
```

## 1.8 R を使った処理の実例紹介

### ■ 「学部の違う二つのクラスで同じ試験を実施した。学部によって成績に差があるか？」

1. エクセルにあるデータをタブ区切りのテキストで保存。( )
2. データを読み込む。
  - ・データフレーム名 <- ((), header = T)

```
classes <- read.table(choose.files(), header = T)
```

### 1. t 検定

- ・ 差がないと仮定する（差がないことがあり得るか）
- ・ クラスごと別々の受講生のデータ（（ ））

```
t.test(classes$classA, classes$classB)
```

- ・ "Welch Two Sample t-test"（等分散でなくてもよい）

## ■ 「ある教材で学習したら成績が上がるか？：事前テストより事後テストの方が良い成績か？」

1. エクセルにあるデータをタブ区切りのテキストで保存。（テキスト（タブ区切り））
2. データを読み込む。
  - ・ データフレーム名 <- ( ), header = T)

```
prepost <- read.table(choose.files(), header = T)
```

### 1. t 検定

- ・ 差がないと想定する（差がないことがあり得るか）
- ・ それぞれの学生ごとにデータがある（ ）：

```
t.test(prepost$pre, prepost$post, paired=T)
```

## 1.9 用意されたデータを使った演習（自分でやる）

### ■ サンプルデータ

1. 学部別クラスデータ
2. 事前・事後テスト

---

### ■ 7月9日の課題の提出：

- ・ 来週 7 月 16 日木曜日授業の初めに
- ・ A4 用紙に印刷して提出
- ・ 1 枚か 2 枚程度
- ・ 二種類の t 検定を行い報告する。
  - ・ なにを、どうして、どのようにしたか。
  - ・ その結果はどうだったか。
  - ・ それから何が考察できるか。
- ・

---

### ■ t 検定のまとめ

t.test

## 2. R による統計分析例

R.intro2015b